LECTURE 3 : IMAGE FEATURES

Motivation : We have seen in Viola-Jones' face detection, one important components is image features : Haar-like features.

↓

Image features are important, because original pixel intensity values are not reliable (= not consistent enough). They are easily affected by changes in light, occlusions, viewing points, etc.

There are two other important features : HoG & SIFT.

[1] Histogram of Oriented Gradients (HoG)

Q : What is gradient?

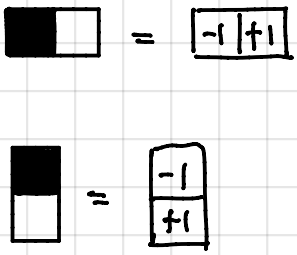
A : Image gradient is basically image derivative, which is expressed as follows.

Derivative :

(1) $\frac{\partial I(x,y)}{\partial x} = I(x+1,y) - I(x,y)$

(2) $\frac{\partial I(x,y)}{\partial y} = I(x,y+1) - I(x,y)$

Convolution :



Q : What is oriented gradient?

A : One pixel has two gradient values : $\frac{\partial I(x,y)}{\partial x}$ & $\frac{\partial I(x,y)}{\partial y}$.

Thus, a pixel gradient is a vector, and a vector has orientation and magnitude.

Orientation : $\tan\theta = \frac{\partial I/\partial y}{\partial I/\partial x} = \frac{I_y}{I_x}$; $\theta = \tan^{-1}\left(\frac{I_y}{I_x}\right)$

Magnitude : $m = \sqrt{I_y^2 + I_x^2}$

Notes on the orientation (θ):

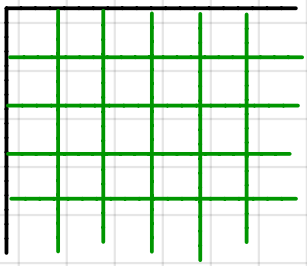
1. θ will be the same when $(I_y=1, I_x=1)$ and $(I_y=-1, I_x=-1)$.
   Hence, dealing with this, we define:

$$\theta = \tan^{-1}\left(\frac{|I_y|}{|I_x|}\right)$$; and keep the signs of I_x & I_y.

   If both are positive, then we consider θ in quadrant 1, and otherwise depending on the signs, it can be all other quadrants.
2. Hence, $0^\circ \leq \theta \leq 360^\circ$.

Q: How can we obtain the histogram of the oriented gradients?
A: Two important concepts: ① Block operation, ② Histogram.

① Block Operation: Divide the input image into blocks:



Each block can be 4x4 or 16x16 pixels
(The choice depends on the image size & applications)

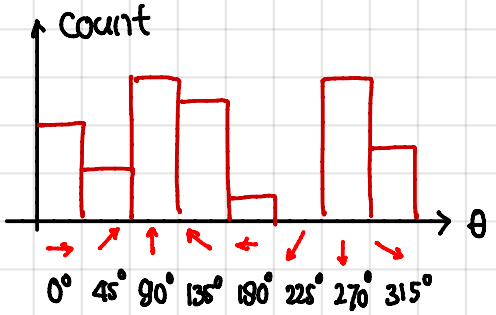
⇓

For each pixel in the block we will have one set of oriented gradient information: (θ_i, m_i)

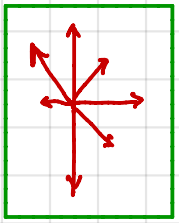
If we have 4x4 pixels (= 16 pixels), then we have 16 sets of (θ_i, m_i), where some of them might have the same/similar values.

② Histogram of Oriented Gradients

1. For each block (cell), we can group the gradients (16 of them) using a histogram with 8 bins (= 8 groups of orientations)

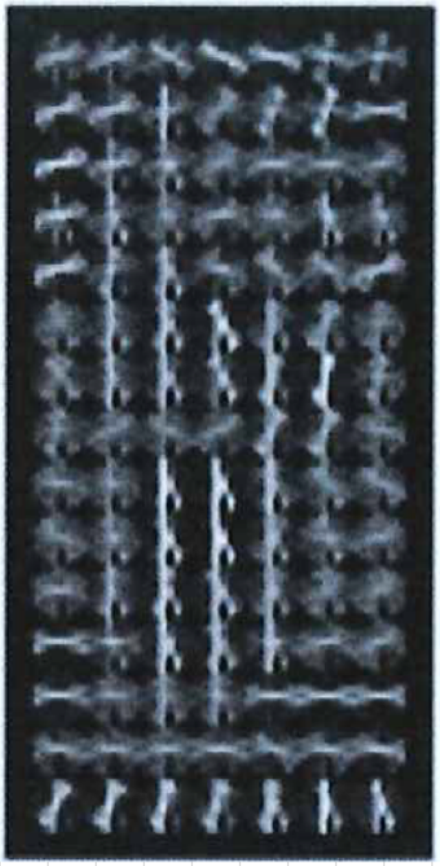


The magnitude influences the height of each bin.



A HoG descriptor: concatenates the histograms of all blocks (or cells) in one long vector!

Input image



HoG

Dominant HoG

• SIFT

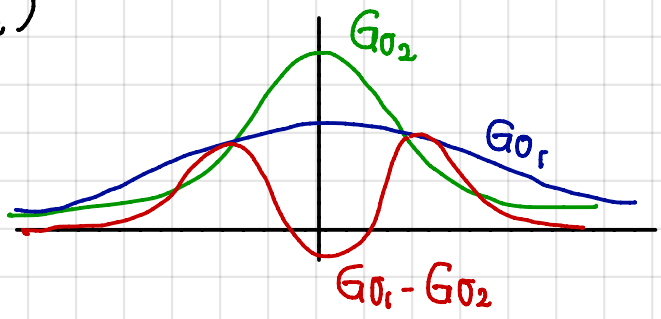
For the complete discussion, see the corresponding slides & paper.

[1] Difference of Gradients (DoG) vs. Laplacian of Gradients (LoG)

» $DoG(I) = (I * G_{\sigma_1}) - (I * G_{\sigma_2})$

$= I * (G_{\sigma_1} - G_{\sigma_2})$

If I is a one-dimensional signal:



» LoG:

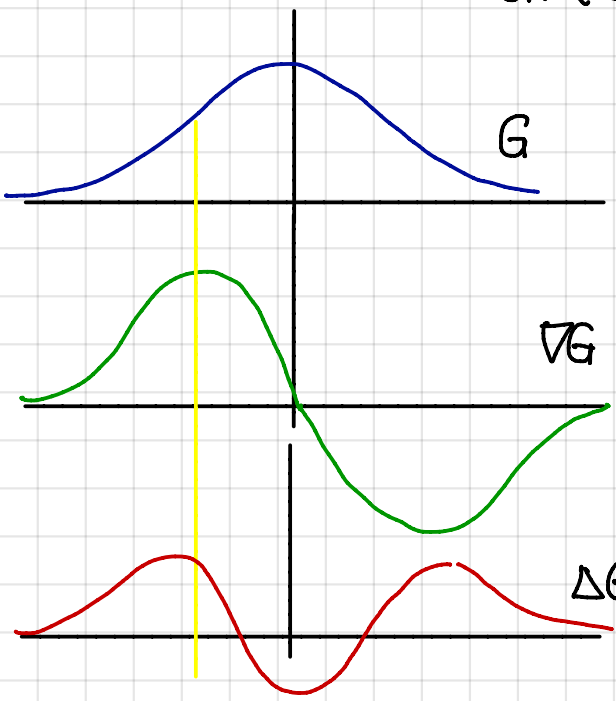
$$\Delta G = \nabla^2 G = \nabla \cdot \nabla G$$

∇ = divergence ∇G = gradient

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right) \qquad \nabla G = \left(\frac{\partial G}{\partial x}, \frac{\partial G}{\partial y}\right)$$

} in 2D

$$\Delta G = \nabla \cdot \nabla G = \frac{\partial}{\partial x}\left(\frac{\partial G}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\partial G}{\partial y}\right) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$



$LoG(I) = I * \Delta G$

DoG is the approximation of LoG.

$DoG(I) \approx LoG(I)$

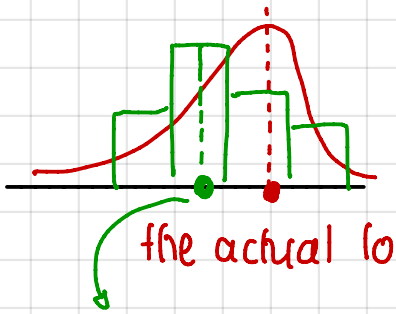
[2] SIFT: Low Contrast Rejection

Motivation:

(1) Due to noise, the actual location of an extremum might be shifted.
(2) Even if we know the actual location, it can be not high enough (= low contrast).

Q: What does it mean by the actual location of an extremum? How to calculate the actual location?

A:



- The green bars are the noisy pixel intensity values
- The red line is the actual (non-noisy) signal

the predicted extremum location (using DoG)

» The actual signal is shifted by noise and discretization.

$\mathbb{D}$ = a 3x3 pixel patch where a keypoint located in the middle, and $\mathbb{D}$ is taken from one of the DoG images.
This 3x3 patch might be affected by noise.

$D(\bar{x})$ = the actual signal ; $\bar{x} = (x, y)$ → the actual location

Unfortunately, we don't know the actual signal location ($\bar{x}$).

Q: How to get $D(\bar{x})$, the signal we want to recover?
A: Through Taylor expansion:

$$\underbrace{f(x)}_{\text{the unknown function}} = f(a) + f'(a)(x-a) + \frac{f''(x-a)}{2!}(x-a)^2 + \dots$$

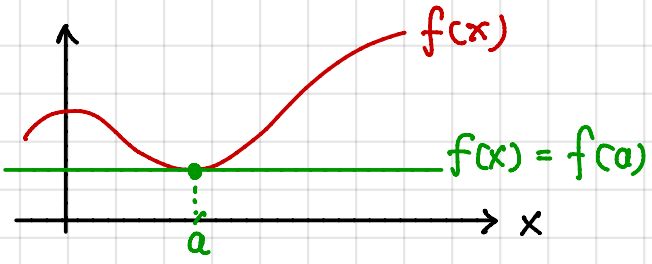
[3] Taylor Expansion

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(x-a)}{2!}(x-a)^2 + \dots$$

Meaning:

1. When $x = a$: $f(x) = f(a)$

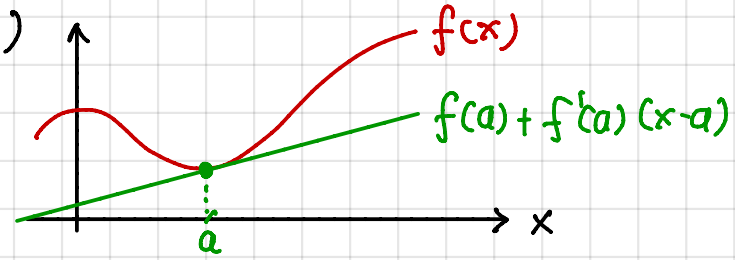
   The approximation is rough & basic



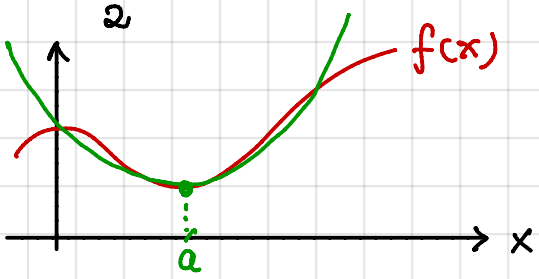
2. When $f(x) = f(a) + f'(a)(x-a)$

   scalar values

   The approximation gets better.



3. When $f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2$



Notes:

- The approximation gets better when we include the higher order functions.
- 'a' is the point where we want to focus our approximation.

[4] Location of Extrema

Using Taylor expansion in 2D:

$$D(\bar{x}) = \mathbb{D}(0) + \frac{\partial \mathbb{D}^T(\bar{x})}{\partial \bar{x}}\bigg|_{\bar{x}=0} \bar{x} + \bar{x}^T \frac{\partial^2 \mathbb{D}(\bar{x})}{2\,\partial \bar{x}^2}\bigg|_{\bar{x}=0} \bar{x}$$

1×1 1×1 1×2 2×1 1×2 2×2 2×1

Note: $\mathbb{D}(0)$

$\mathbb{D} =$

(-1, 0, +1 on both axes)

;

$$\frac{\partial \mathbb{D}(\bar{x})}{\partial \bar{x}}\bigg|_{\bar{x}=0} = \begin{bmatrix} \partial \mathbb{D}/\partial x \\ \partial \mathbb{D}/\partial y \end{bmatrix}\bigg|_{\bar{x}=0}$$

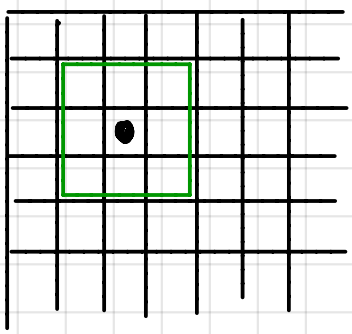
$$\frac{\partial^2 \mathbb{D}(\bar{x})}{\partial \bar{x}^2}\bigg|_{\bar{x}=0} = \begin{bmatrix} \frac{\partial^2 \mathbb{D}}{\partial x^2} & \frac{\partial^2 \mathbb{D}}{\partial x \partial y} \\ \frac{\partial^2 \mathbb{D}}{\partial y \partial x} & \frac{\partial^2 \mathbb{D}}{\partial y^2} \end{bmatrix}\bigg|_{\bar{x}=0}$$

Q: What are these $\mathbb{D}$, $\mathbb{D}_x$, $\mathbb{D}_y$, $\mathbb{D}_{xx}$, $\mathbb{D}_{xy}$, $\mathbb{D}_{yy}$?

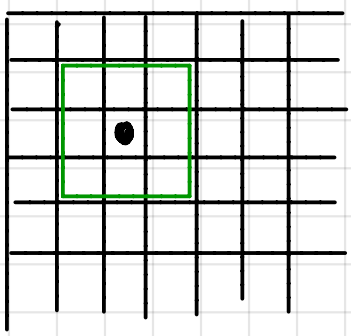
A: $\mathbb{D}$ is a DoG image (the output of the previous step), where there are a number of keypoints. For each of these keypoints, we take 3×3 pixels from $\mathbb{D}$.

$\mathbb{D}_x$ (or $\frac{\partial \mathbb{D}}{\partial x}$) & $\mathbb{D}_y$ (or $\frac{\partial \mathbb{D}}{\partial y}$) are the first derivative image of $\mathbb{D}$.

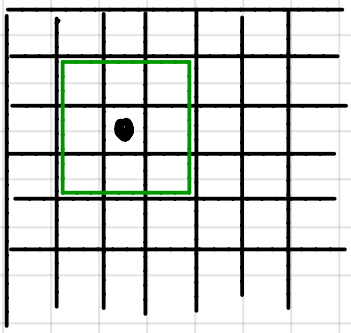
$\mathbb{D}$:



$\mathbb{D}_x$:



$\mathbb{D}_{xx}$:



we compute $\mathbb{D}_y$, $\mathbb{D}_{yy}$, and $\mathbb{D}_{xy}$ (which is the same as $\mathbb{D}_{yx}$) images in the same way.

To find the true extremum means:

$$\frac{\partial D(\bar{x})}{\partial \bar{x}} = 0 \quad \rightarrow \text{the output is a vector of } 2\times 1.$$

$$\frac{\partial D(0)}{\partial \bar{x}} + \frac{\partial}{\partial \bar{x}}\left[\left.\frac{\partial \mathbb{D}^T}{\partial \bar{x}}\right|_{\bar{x}=0} \bar{x}\right] + \frac{\partial}{\partial \bar{x}}\left[\bar{x}^T \left.\frac{\partial^2 \mathbb{D}}{2\,\partial \bar{x}^2}\right|_{\bar{x}=0} \bar{x}\right] = 0$$

(the bracketed term is constant)

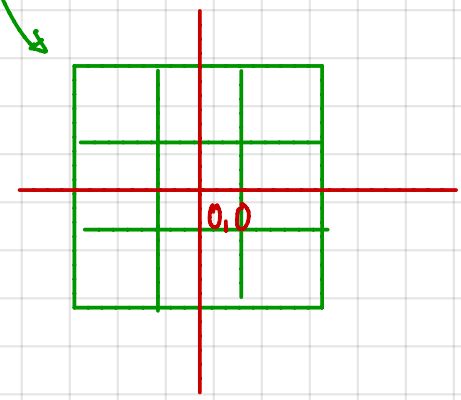
$$0 + \left.\frac{\partial \mathbb{D}(\bar{x})}{\partial \bar{x}}\right|_{\bar{x}=0} + \left.\frac{\partial^2 \mathbb{D}(\bar{x})}{\partial \bar{x}^2}\right|_{\bar{x}=0} \bar{x} = 0$$

$$\bar{x}^* = -\left(\left.\frac{\partial^2 \mathbb{D}(\bar{x})}{\partial \bar{x}^2}\right|_{\bar{x}=0}\right)^{-1} \left.\frac{\partial \mathbb{D}(\bar{x})}{\partial \bar{x}}\right|_{\bar{x}=0}$$

If the offset of $\bar{x}^* > 0.5$

then: check the contrast based on the new location $\bar{x}^* \rightarrow \mathbb{D}(\bar{x}^*)$

else check the contrast based on $\mathbb{D}(0)$.



the distance from (0,0) to the cell boundaries is 0.5.

[5] Low Contrast : Checking / Verification

Based on the new location $\bar{x}^*$, we check if the extremum is high enough:

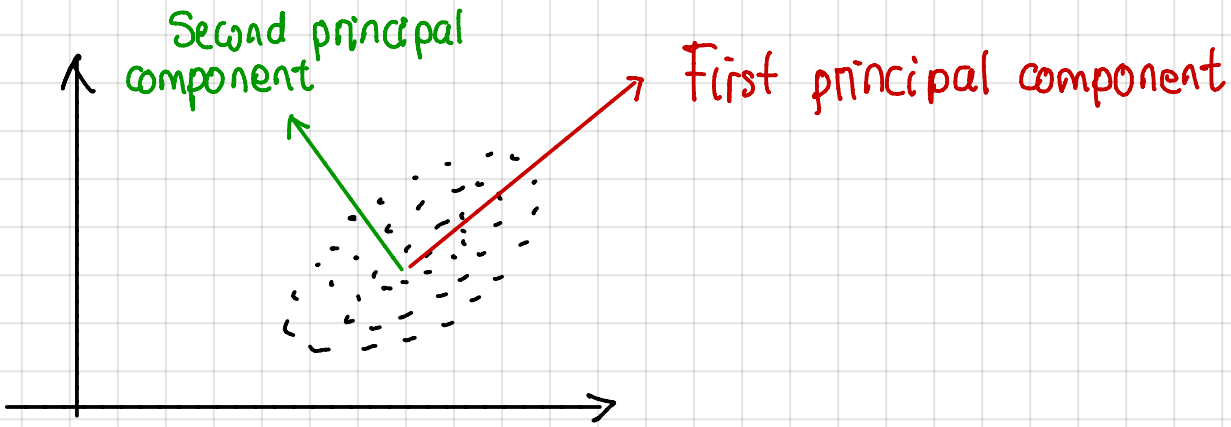
$$D(\bar{x}^*) = \underset{1\times1}{D[0]} + \underset{1\times2}{\frac{\partial D^T(\bar{x})}{\partial \bar{x}}\Big|_{\bar{x}=0}} \underset{2\times1}{\bar{x}^*}$$

If $|D(\bar{x}^*)| < 0.03$ then reject!

if assumes the image intensity of the input is between 0 ~ 1 (instead of 0 ~ 255).

---

[6] First and Second Principal Components of Curvatures



α = The eigen value of the first principal component

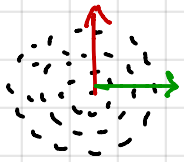
β = The eigen value of the second principal component

$\alpha \geq \beta$

[7] Corner Detection

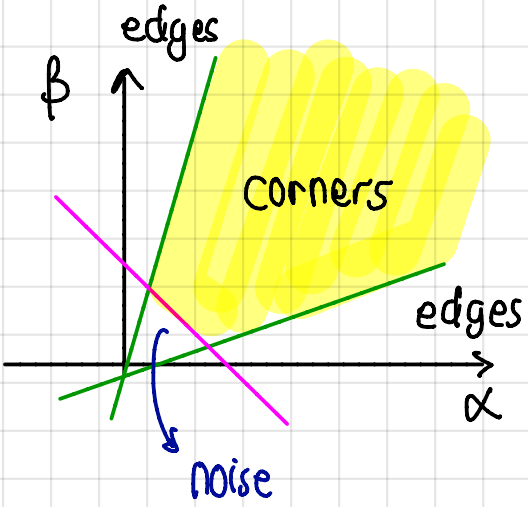
Eigenvalues (α & β) can indicate whether a pixel cluster is edge or corner:

$\alpha \gg \beta \rightarrow$ edge



$\alpha \approx \beta \rightarrow$ corner

$\Rightarrow$



Unlike the above illustration, our data is only 3x3 pixels (9 pixels), which is too sparse to calculate the principal axes.

↓

Solution: to use the ratio of α & β:

1. Hessian matrix: $H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$

2. $Tr(H) = D_{xx} + D_{yy} = \alpha + \beta$

   $Det(H) = D_{xx} D_{yy} - D_{xy}^2 = \alpha\beta$

3. $\frac{Tr^2(H)}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta}$; define: $\alpha = r\beta$

   $= (r\beta+\beta)^2 / r\beta^2 = \frac{(r+1)^2}{r}$

if $\alpha = \beta \rightarrow r = 1$: $Tr^2(H)/Det(H) = 4$ → corner

if $\alpha = 2\beta \rightarrow r = 2$: $Tr^2(H)/Det(H) = 9/2 = 4.5$

if $\alpha = 10\beta \rightarrow r = 10$: $Tr^2(H)/Det(H) = 121/10 = 12.1$ → edge

Therefore: if $Tr^2(H)/Det(H) < 12.1$

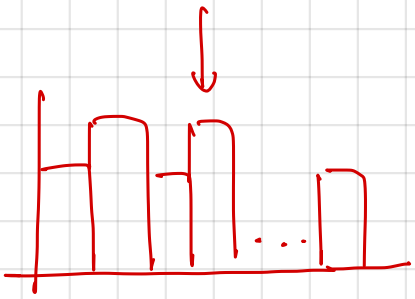
then retain the keypoints, else reject them.

[8] SIFT: Generating Descriptors

(1) Orientation assignment

Goal: to make the descriptor invariant to image rotation.

1. Extract 16x16 pixels surrounding a keypoint.
2. Create a histogram of orientations with 36 bins covering 360°.
3. Choose the highest peak in the histogram, and any peaks above 80%, to calculate the orientation normalization.

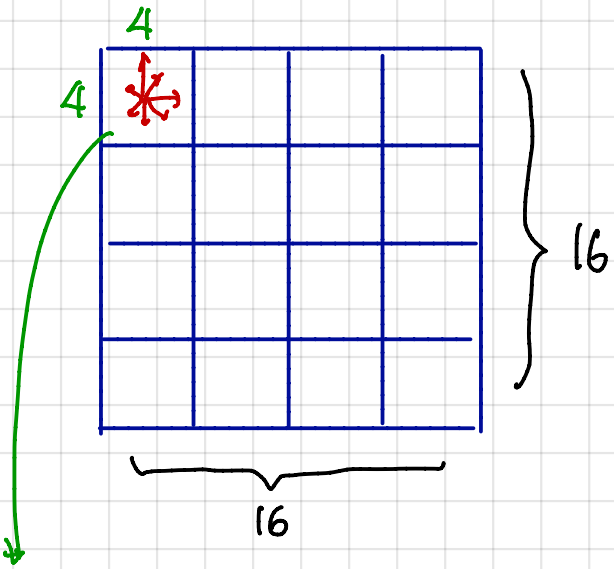


Besides multiplying with the magnitudes, we also multiply with the Gassian-weighted circular window, so that the center (keypoint) gets more weight than the peripheries.

(2) Descriptor:

One keypoint generates one descriptor, which has a length of 128.

These 128 numbers are obtained from 16x16 pixels:



For each block, we compute the histogram of gradients with 8 bins (= orientations)

Hence, for 1 block of 4x4 pixels we have 8 numbers. Thus, in total we have:

$$4 \times 4 \times 8 = 128.$$